TOWARDS THE OPTIMAL BAYES CLASSIFIER USING AN EXTENDED SELF-ORGANISING MAP

Hujun Yin and Nigel M. Allinson

Department of Electronics, University of York
York, YO1 5DD, UK
E-mail: hy@ohm.york.ac.uk
Tel: (+44) (01904) 433221, Fax: (+44)(01904) 433222

Abstract: *In this paper, we propose an extended self-organising learning scheme, in which both distance measure and neighbourhood function have been replaced by the neuron's posterior probabilities. Updating of weights is within a limited but fixed sized neighbourhood of the winner. Each unit will converge to one component of a mixture distribution of input samples, so that an optimal pattern classifier can be formed. The proposed learning scheme can be used to train other forms of unsupervised networks, such as radial-basis-function networks. An application example on textured image segmentation is presented.*

INTRODUCTION

The self-organising map (SOM) has been widely used in data clustering, pattern classification, and vector quantisation (VQ) [Kohonen 1990]. The SOM is a biologically inspired unsupervised learning algorithm. It uses a two-dimensional cortex-like array of neurons to capture and represent input data from a frequently higher dimension space, and then form a map with certain topologically preserved properties. The SOM has two-fold characteristics or potential uses: quantising the input space and classifying input data into underlying patterns. However the SOM is more likely to be optimal when used as a vector quantiser than used as a classifier, because it tends to approximate the input data space. The authors have proved the SOM's convergence for arbitrary dimensional cases, and that it will eventually satisfy the two necessary conditions for VQ [Yin and Allinson 1995].

When used as a classifier, a conventional SOM normally will not form optimal Bayesian classification unless the input data are uniformly distributed or patterns are well separated. In most cases, the pattern distributions are overlapped, and their joint distribution can be described by a mixture distribution (MD). Like the *k*-means algorithm, the SOM often results in more classification errors than the Bayes classifier. To form a Bayesian classification, the SOM needs some form of supervision to label and adjust a pre-turned SOM (its weights) as in the LVQ1, LVQ2, and LVQ3 [Kohonen 1990], or another linear supervised layer [Haykin 1994]. However if the unsupervised training phase can accurately capture the underlying MD, it will assist the supervised training phase, or it can be used to form an unsupervised Bayesian classifier.

MIXTURE DISTRIBUTION AND UNSUPERVISED LEARNING

Mixture Distribution Model

The mixture distribution models can be seen in many practical pattern classification applications. Each sample, **x,** from a d_1-dimensional input space, $\mathbf{X} \in \Re^{d_1}$, is to be assigned to one of N distinct

classes, ω_1, ω_2, ...ω_N, each of which has a prior probability. In each pattern class samples are distributed according to a prescribed class-conditional probability density. The joint-probability-density of the samples is a MD, given by [Duda and Hart 1973, Everitt and Hand 1981],

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{N} p(\mathbf{x}|\omega_i, \theta_i) P(\omega_i) \tag{1}$$

where $p(\mathbf{x}|\omega_i, \theta_i)$ is the *i*-th class-conditional density, and θ_i are the sufficient statistics, or parameter vector, for the *i*-th class-conditional density, *i*=1, 2, ... *N*. $\Theta=(\theta_1, \theta_2, ... \theta_N)^T$. $P(\omega_i)$ is the prior probability of the *i*-th class and is sometimes called the mixing parameters, or mixing weights.

For most unsupervised learning applications, only the number of classes and their class-conditional density forms are known; the other parameters have to be learnt unsupervised from a set of *n* unlabelled independent samples, $\chi=\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n\}$. In this case, maximising the joint-likelihood (ML) of all observed samples, $p\{\chi|\Theta) = \prod p(\mathbf{x}_k|\Theta)$, may lead to a singular solution. When restricted to the largest finite maxima and Gaussian components, it results in the following implicit equations for the parameters [Duda and Hart 1973]:

$$\hat{\bar{\mu}}_i = \frac{\sum \hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)\mathbf{x}_k}{\sum \hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)} \tag{2}$$

$$\hat{\bar{\sigma}}_i = \frac{\sum \hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)(\mathbf{x}_k - \hat{\bar{\mu}}_i)(\mathbf{x}_k - \hat{\bar{\mu}}_i)^T}{\sum \hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)} \tag{3}$$

$$\hat{P}(\omega_i) = \frac{1}{n}\sum \hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i) \tag{4}$$

$$\hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i) = \frac{p(\mathbf{x}_k|\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{\sum_{j=1}^{N} p(\mathbf{x}_k|\omega_j, \hat{\theta}_j)\hat{P}(\omega_j)} \tag{5}$$

These equations can only be solved by using some nonlinear optimisation methods. For example, the expectation-maximisation method has been used to obtain an iterative procedure for these parameters [Tarassenko and Roberts 1994]. Normally they require very intensive computation in order to achieve a solution. Only local optima can be guaranteed, and results depends on the initial states.

Maximum Likelihood Competitive Learning

Nowlan (1990) has proposed two possible modifications to the winner-take-all (WTA) learning for MD problems. One is *maximum-likelihood competitive learning* (MLCL), in which the winner, *v*, is selected according to its weighted likelihood function instead of Euclidean distance, i.e.

$$v = \arg\max_i \hat{P}(\omega_i) p(\mathbf{x}_k|\omega_i, \hat{\theta}_i) \tag{6}$$

This is achieved by choosing the maximum component as a better approximation to the MD at $\mathbf{x}_k$ than choosing any others. This is true only when components are well separated, or at worst slightly overlapped. The ML learning gives better clustering results than the simple Euclidean distance

WTA algorithm. When the components of the mixture are Gaussian with equal prior probabilities, the Mahalanobis distance measure is equal to the ML measure.

The other is "*soft*" *competitive learning*, in which neurons share responsibility in proportion to their posterior probabilities. In "soft" competitive learning or sharing schemes, all other neurons in addition to the winner have to be updated. This may correspondingly increase the computation costs when it is not implemented in parallel, especially when the number of neurons is very large.

Probabilistic WTA Learning

Osman and Fahmy (1994) have proposed a so-called *probabilistic* WTA, in which the winner is chosen probabilistically according to the neurons' posterior probabilities, i.e. eqn. (5), to avoid updating all the neurons' weights. This will increase the number of total learning iterations, because each sample has to be input very many times in order that all possible units learn.

AN EXTENDED SELF-ORGANISING LEARNING FOR BAYESIAN CLASSIFICATION

In this section, the SOM algorithm is extended and applied to the kernel learning networks for MD. The network places *N* units in the input dimensional space or a reduced dimension space. Each unit is a Gaussian kernel, with its mean vector, $\hat{\bar{\mu}}_i$, covariance matrix, $\hat{\bar{\sigma}}_i$, and mixing weight, $\hat{P}(\omega_i)$, *or* $\hat{P}_i$, as learning parameters, or self-organising learning weights. At each time step, *t*, a sample, denoted by $\mathbf{x}_k(t)$, is randomly taken from χ. The winner is chosen according to its kernel output, i.e. estimated posterior probability, as in eqn. (6). Then within a neighbourhood of the winner, η_v, the weights are updated according to the following rules:

$$\hat{\bar{\mu}}_i(t+1) = \hat{\bar{\mu}}_i(t) + \alpha(t)\hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)(\mathbf{x}_k(t) - \hat{\bar{\mu}}_i(t)) \tag{7}$$

$$\hat{\bar{\sigma}}_i(t+1) = \hat{\bar{\sigma}}_i(t) + \alpha(t)\hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i)\{(\mathbf{x}_k(t) - \hat{\bar{\mu}}_i(t))(\mathbf{x}_k(t) - \hat{\bar{\mu}}_i(t))^T - \hat{\bar{\sigma}}_i(t)\} \tag{8}$$

$$\hat{P}_i(t+1) = \hat{P}_i(t) + \alpha(t)(\hat{P}(\omega_i|\mathbf{x}_k, \hat{\theta}_i) - \hat{P}_i(t)) \tag{9}$$

where *the adaptive gain*, $\alpha(t)$, is the same as in the original SOM.

As can be seen, the original SOM's neighbourhood functions, which are fixed in shape but shrinking in size, have been replaced by adaptive posterior probability functions. The neighbourhood size, which depends on the covariance of components, should be fixed but large enough. The topological ordering property of the SOM ensures that the posteriors of the components that are outside the neighbourhood are very small. Therefore,

$$p(\mathbf{x}|\hat{\Theta}) \approx \sum_{i \in \eta_v} p(\mathbf{x}|\omega_i, \hat{\theta}_i)\hat{P}(\omega_i) \tag{10}$$

The convergence of the algorithm can be proved using a theorem proposed and proved by the authors [Yin and Allinson 1995]. The learning parameters will eventually converge to the conditions (2)-(5).

APPLICATIONS TO IMAGE SEGMENTATION

The algorithm has been applied to a real problem, unsupervised segmentation of textured image. The whole network structure is similar to the hierarchical neural structure, previously described in

[Yin and Allinson 1994a, 1994b]. There are two hierarchical self-organising layers, one is the *estimating layer*, which is a SOM chain, while the other is the *fine segmenting layer*, which is a simplified two dimensional SOM array (a local voting network in this example). The initial parameters of the networks are chosen in random. The test images are of 128×128 in size, and are composite of two texture regions. At each iteration, a randomly moving window, which is also shrinking with time (large at the beginning, say 70×70, and small later, say 10×10), locates an area. A least-square estimator is used to obtain crude parameters of the Markov random field (MRF) model for this area (a second-order MRF model is used). Then the estimating layer learns to classify the parameters and estimate the real parameters for each texture region. The winning neuron yields outputs to the local voting layer in that area. After many iterations a fine segregated picture can be obtained, and the weights of the estimating layer will be the MRF model parameters of each region. In present experiment, the original SOM algorithm in the estimating layer has been replaced by the proposed extended SOM algorithm so that to obtain a better estimate of the underlying patterns' MD. Improved results have been achieved, especially in the estimating layer, as can be seen in Fig. 1 (comparing (b) with (d), and (g) with (i) in Fig. 1). This implies that the proposed algorithm can give a better interpretation of the sample distribution.

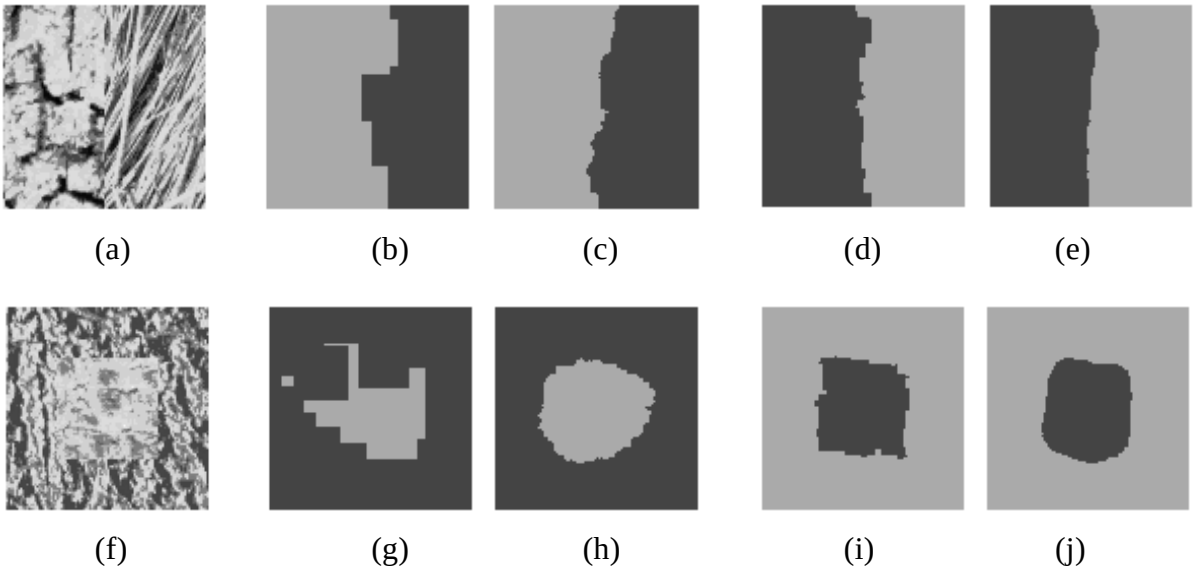

Fig. 1 Textured Image Segmentation
(a), (f) are composite Brodatz texture images with size of 128×128 pixels. (b) and (c), (g) and (h) are previous results; (d) and (e), (i) and (j) are new results. (b), (d), (g), and, (i) are the outputs of the estimating layer; (c), (e), (h), and, (j) are the outputs of the whole system.

CONCLUSIONS

An extended self-organising learning algorithm has been proposed. The normal neighbourhood functions and distance measures have been replaced by the neuron's posterior probabilities. The algorithm can be used in unsupervised kernel-like learning to estimate the underlying density modelled by a mixture of overlapped components. Like the SOM, the extended algorithm is a simple algorithm and easy to implement. Its neighbourhood function can form a topologically ordered map, which may provide high noise-tolerance for VQ, and makes local learning possible. It also can overcome under-utilisation or singular problems. Since it is generally not an exact gradient descent method, and is independent of initial states, it may have a higher possibility of escaping local minima. Effective monitoring of its learning procedures, or using of an on-line validation program, may be helpful in forming a globally optimal map.

REFERENCES

Duda, R. O. and Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley.

Everitt, B. S. and Hand, D. L. (1981), *Finite Mixture Distributions*, London: Chapman and Hall.

Haykin, S. (1994), *Neural Networks, a Comprehensive Foundation*, New York: Macmillan College Publishing.

Kohonen, T. (1990), "The self-organising map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464-1480.

Nowlan, S. J. (1990), "Maximum likelihood competitive learning," In *Advances in Neural Information Processing Systems 2*, Tourezky, D. S. ed., pp. 574-582.

Osman, H. and Fahmy, M. M. (1994), "Probabilistic winner-take-all learning algorithm for radial-basis-function neural classifiers," *Neural Computation*, vol. 6, no. 5, pp. 927-943.

Tarassenko, L. and Roberts, S. (1994), "Supervised and unsupervised learning in radial-basis-function classifiers," *IEE Proc. Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 201-216.

Yin, H. and Allinson, N. M. (1994a), "Self-organised segmentation for textured images," In *Proc. ICANN'94*, pp. 1149-1152.

Yin, H. and Allinson, N. M. (1994b), "Unsupervised segmentation of textured images using a hierarchical neural structure," *Electronics Letters*, vol. 30, no. 22, pp. 1842-1843.

Yin, H. and Allinson, N. M. (1995), "On the distribution and convergence of feature space in self-organising maps," To appear in *Neural Computation*, vol. 7, no. 6. pp. 1154-1163.